


# OTIMIZAÇÃO E BALANCEAMENTO DE BANCO DE DADOS

Lílian Simão Oliveira

- 
- Material baseado no livro: Sistemas de Gerenciamento de Banco de Dados – Ramakrishnan – Gehrke
  - Slides do Prof. Ricardo Rocha

# Tuning



- Tuning pode ser dividido em 3 grandes tipos:
  - ▣ Refinamento do esquema das relações e as consultas/atualizações no Banco de Dados
  - ▣ Configuração no sistema operacional
  - ▣ Configuração dos parâmetros dos SGBD's

# Refinamento de Esquema



- Problemas causados pela redundância
  - Armazenamento redundante
  - Anomalias de atualização
  - Anomalias de inserção
  - Anomalias de exclusão

# Regras para um bom desempenho

- **Regra 1:** Os atributos de uma relação devem representar apenas uma entidade ou um relacionamento.
  - Atributos de entidades ou relacionamentos diferentes devem estar separados o mais possível. Apenas chaves externas devem ser usadas para referenciar outras relações.
  - É mais fácil explicar o significado de uma relação se esta representar apenas uma entidade ou relacionamento. Evita ambiguidades no significado das relações.
- Exemplo de uma boa relação do ponto de vista lógico mas que viola a regra 1:
  - EMP\_DEP(NomeEmp, NumBI, Endereço, DataNasc, NumDep, NomeDep, GerenteBI)
- Problemas com a relação EMP\_DEP:
  - Os valores dos atributos NomeDep e GerenteBI aparecem repetidos para os empregados que trabalham num mesmo departamento.

# Regras para um bom desempenho

- **Regra 2:** Evitar a possibilidade de ocorrerem anomalias nas operações de inserção, remoção ou alteração.
  - Se por razões de eficiência isso não for possível, garantir que os utilizadores/programas que manipulam a BD conhecem essas anomalias e as evitam.
- Exemplo de anomalias de inserção em EMP\_DEP:
  - Não é possível inserir um novo departamento a menos que seja associado a um empregado.
  - Ao inserir um empregado é necessário garantir que os valores dos atributos NomeDep e GerenteBI são consistentes com os dos restantes empregados desse departamento.
- Exemplo de anomalias de remoção em EMP\_DEP:
  - Se removermos o último empregado para um determinado departamento, então a informação desse departamento também é removida.
- Exemplo de anomalias de alteração em EMP\_DEP:
  - A alteração do nome de um departamento leva a que essa alteração tenha que ser feita sobre todos os tuplos dos empregados que nele trabalham.

# Regras para um bom desempenho

- **Regra 3:** Evitar atributos que possam ter valores NULL numa grande parte dos tuplos duma relação.
  - Colocar esse tipo de atributos em relações separadas juntamente com a chave primária.
  - Minimiza o espaço necessário para guardar os tuplos da relação e evita problemas no cálculo de funções de agregações sobre esses atributos.
- **Exemplo:**
  - Se apenas 5% dos empregados tiverem gabinete individual não faz sentido incluir um atributo NumGabinete na relação EMP\_DEP. Uma melhor solução é criar uma relação GABINETE(EmpBI, NumGabinete) para guardar essa informação.

# Regras para um bom desempenho

- **Regra 4:** Evitar relações que tenham atributos relacionados que não sejam combinações do tipo chave externa com chave primária.
  - Operações de junção sobre esses atributos poderão originar tuplos falsos. Não verificam a **propriedade de junção-não-aditiva** (ou junção-sem-perdas de informação).
- Considere as seguintes relações e assuma que a localização de cada projecto é única, ou seja, dois projectos diferentes têm sempre localizações diferentes:
  - TRAB\_PROJ(EmpBI, NumProj, Horas, NomeProj, LocalizaçãoProj)
  - EMP\_LOC(NomeEmp, LocalizaçãoProj)
- Problemas com as relações TRAB\_PROJ e EMP\_LOC:
  - A operação de junção natural TRAB\_PROJ \* EMP\_LOC dá origem a mais tuplos do que aqueles que seriam obtidos pela junção das tabelas originais TRABALHA\_EM, PROJECTO e EMPREGADO. Note que isto acontece mesmo com localizações únicas para cada projecto.



# funcionarios

| cpf        | nome  | cargo    | Salário  |
|------------|-------|----------|----------|
| 123-22-366 | João  | Técnico  | 800,00   |
| 231-11-777 | Maria | Gerente  | 1.000,00 |
| 142-88-976 | Joana | Vendedor | 900,00   |
| 421-66-891 | José  | Técnico  | 800,00   |

# funcionarios

| cpf        | nome  | cargo    | Salário  |
|------------|-------|----------|----------|
| 123-22-366 | João  | Técnico  | 800,00   |
| 231-11-777 | Maria | Gerente  | 1.000,00 |
| 142-88-976 | Joana | Vendedor | 900,00   |
| 421-66-891 | José  | Técnico  | 800,00   |

Redundância

- ❑ Problemas:
  - ❑ Armazenamento redundante
  - ❑ Anomalias de atualização
  - ❑ Anomalias de inserção
  - ❑ Anomalias de exclusão

# Decomposição

Uma decomposição de um esquema relacional  $R$  consiste na substituição do esquema em questão por dois ou mais esquemas relacionais tais que:

1. cada um contenha um subconjunto de atributos  $R$ ;
2. juntos incluam todos os atributos de  $R$ .

funcionarios

| cpf        | nome  | cargo    |
|------------|-------|----------|
| 123-22-366 | João  | Técnico  |
| 231-11-777 | Maria | Gerente  |
| 142-88-976 | Joana | Vendedor |
| 421-66-891 | José  | Técnico  |

cargos

| cargo    | Salário  |
|----------|----------|
| Técnico  | 800,00   |
| Gerente  | 1.000,00 |
| Vendedor | 900,00   |

# Decomposição



- Vantagens – facilita a manutenção dos dados, evitando inconsistências e redundâncias
  
- Desvantagem – pode-se criar mais problemas que resolvê-los. Deve-se observar:
  - ▣ Se o esquema de relação está em uma das formas normais;
  - ▣ Propriedade de Junção sem perda (não tem perda de informação)
  - ▣ Preservação da dependência

# Formas Normais



- 1FN
- 2FN
- 3FN
- Forma Normal de Boyce-Codd (FNBC)

# 1FN – 1ª Forma Normal

- Um esquema relacional está na primeira forma normal se todos os atributos forem atômicos (não divisíveis)
- Normalização 1NF
  - ▣ Decompor atributos compostos em atributos atômicos
  - ▣ Decompor atributos multi-valorado em relação com chave externa:
    - A relação DEPARTAMENTO(Nome, Num, {Localização}) pode ser decomposta em DEPARTAMENTO(Nome, Num, Localização), mas a melhor solução: DEPARTAMENTO(Nome, Num) e Localização(NumDep, Localização)

# Dependência Funcional

- **Dependência funcional (FD)** é uma restrição entre 2 conjuntos de atributos.
- Seja  $R(A_1, A_2, \dots, A_n)$  um esquema relacional e sejam  $X$  e  $Y$  dois subconjuntos de atributos de  $R$ . Diz-se que  $X$  determina funcionalmente  $Y$  (ou que  $Y$  depende funcionalmente de  $X$ ), representado por  $X \rightarrow Y$ , se quaisquer dois tuplos de  $R$  que têm os mesmos valores em  $X$  também têm os mesmos valores em  $Y$ .

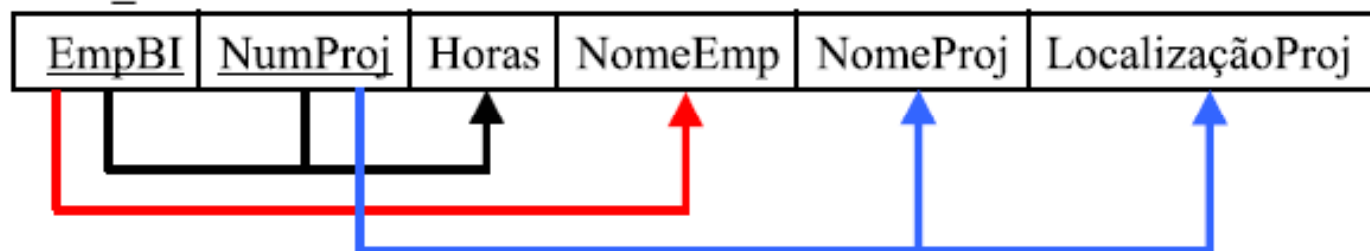
$$X \rightarrow Y \equiv \forall t_1, t_2 \in r(R): t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

- Interpretação:
  - Os valores da componente  $Y$  de tuplos de  $R$  dependem dos valores da componente  $X$ .
  - Os valores da componente  $X$  de tuplos de  $R$  determinam os valores da componente  $Y$ .
- As chaves de uma relação são casos particulares de dependências funcionais. Se  $X$  for uma chave de  $R$  então  $X \rightarrow Y$  para qualquer subconjunto  $Y$  de atributos de  $R$ .
- O facto de  $X \rightarrow Y$  em  $R$  nada permite concluir acerca de  $Y \rightarrow X$  em  $R$ .

# Dependência Funcional

- Considere a seguinte relação:
  - EMP\_PROJ(EmpBI, NumProj, Horas, NomeEmp, NomeProj, LocalizaçãoProj)
- Dependências funcionais da relação EMP\_PROJ:
  - $\text{EmpBI} \rightarrow \text{NomeEmp}$
  - $\text{NumProj} \rightarrow \{\text{NomeProj}, \text{LocalizaçãoProj}\}$
  - $\{\text{EmpBI}, \text{NumProj}\} \rightarrow \text{Horas}$

EMP\_PROJ





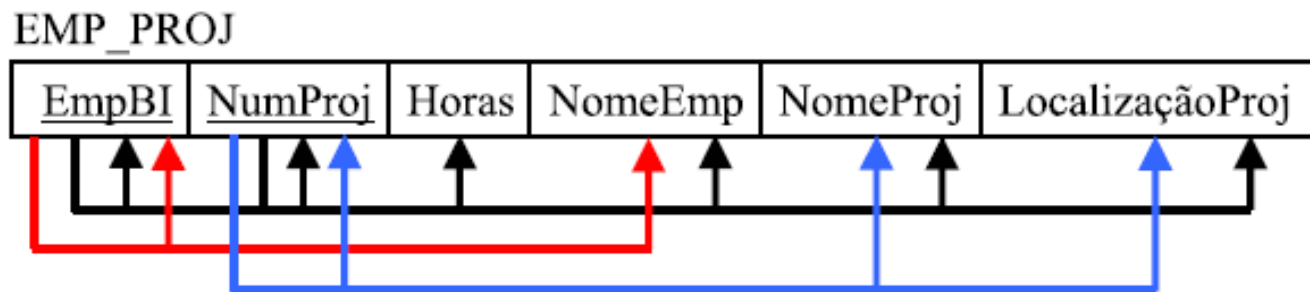
# Dependência Funcional

- Normalmente, a partir de um conjunto de dependências funcionais é possível inferir outras dependências funcionais.
- Regras de inferência de Armstrong (1974):
  - **Regra Reflexiva:** Se  $X \supseteq Y$  então  $X \rightarrow Y$  (ou  $X \rightarrow X$ ).
  - **Regra Aditiva:** Se  $X \rightarrow Y$  então  $XZ \rightarrow YZ$  (ou se  $X \rightarrow Y$  então  $XZ \rightarrow Y$ ).
  - **Regra Transitiva:** Se  $X \rightarrow Y$  e  $Y \rightarrow Z$  então  $X \rightarrow Z$ .
- O **fecho** de um conjunto de dependências funcionais  $F$  é o conjunto  $F^+$  de todas as dependências funcionais que podem ser inferidas a partir de  $F$ .
- O **fecho** de um conjunto de atributos  $X$  de  $F$  é o conjunto  $X^+$  de todas os atributos que podem ser inferidos a partir de  $X$ .
- $F^+$  e  $X^+$  podem ser calculados por aplicação sucessiva das regras de Armstrong.

# Dependência Funcional

## ■ Fecho das dependências funcionais da relação EMP\_PROJ:

- $\{\text{EmpBI}\}^+ \rightarrow \{\text{EmpBI}, \text{NomeEmp}\}$
- $\{\text{NumProj}\}^+ \rightarrow \{\text{NumProj}, \text{NomeProj}, \text{LocalizaçãoProj}\}$
- $\{\text{EmpBI}, \text{NumProj}\}^+ \rightarrow \{\text{EmpBI}, \text{NumProj}, \text{Horas}, \text{NomeEmp}, \text{NomeProj}, \text{LocalizaçãoProj}\}$



# Dependência Funcional

- Uma dependência funcional  $X \rightarrow Y$  diz-se **parcial** se a remoção de algum atributo de  $X$  não deixar de determinar funcionalmente  $Y$ .

$X \rightarrow Y$  é uma dependência funcional parcial se  $\exists A \in X: (X - A) \rightarrow Y$

- Uma dependência funcional  $X \rightarrow Y$  diz-se **completa** (ou **não parcial**) se a remoção de um qualquer atributo de  $X$  deixar de determinar funcionalmente  $Y$ .

$X \rightarrow Y$  é uma dependência funcional completa se  $\forall A \in X: (X - A) \not\rightarrow Y$

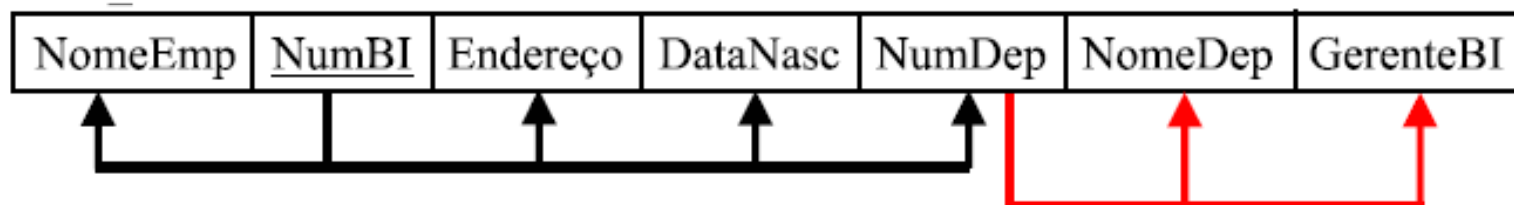
- Exemplos para a relação EMP\_PROJ:

- $\{\text{EmpBI}, \text{NumProj}\} \rightarrow \text{NomeEmp}$  é uma dependência funcional parcial porque  $\text{EmpBI} \rightarrow \text{NomeEmp}$  também se verifica.
- $\{\text{EmpBI}, \text{NumProj}\} \rightarrow \text{Horas}$  é uma dependência funcional completa porque nem  $\text{EmpBI} \rightarrow \text{Horas}$  nem  $\text{NumProj} \rightarrow \text{Horas}$  se verificam.

# Dependência Funcional

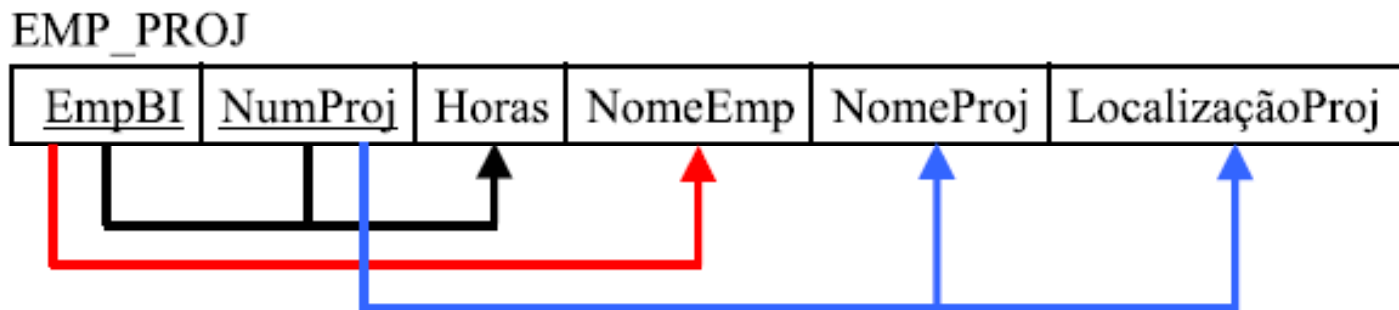
- Uma dependência funcional  $X \rightarrow Y$  diz-se **transitiva** se existir um conjunto de atributos não-chave que depende funcionalmente de  $X$  e determina funcionalmente  $Y$ .  
 $X \rightarrow Y$  é uma dependência funcional transitiva se  $\exists Z$  não-chave:  $X \rightarrow Z$  e  $Z \rightarrow Y$ .
- Exemplo para a relação EMP\_DEP:
  - $\text{NumBI} \rightarrow \{\text{NomeDep}, \text{GerenteBI}\}$  é uma dependência funcional transitiva porque  $\text{NumBI} \rightarrow \text{NumDep}$ ,  $\text{NumDep} \rightarrow \{\text{NomeDep}, \text{GerenteBI}\}$  e  $\text{NumDep}$  não pertence a nenhuma chave candidata de EMP\_DEP.

EMP\_DEP



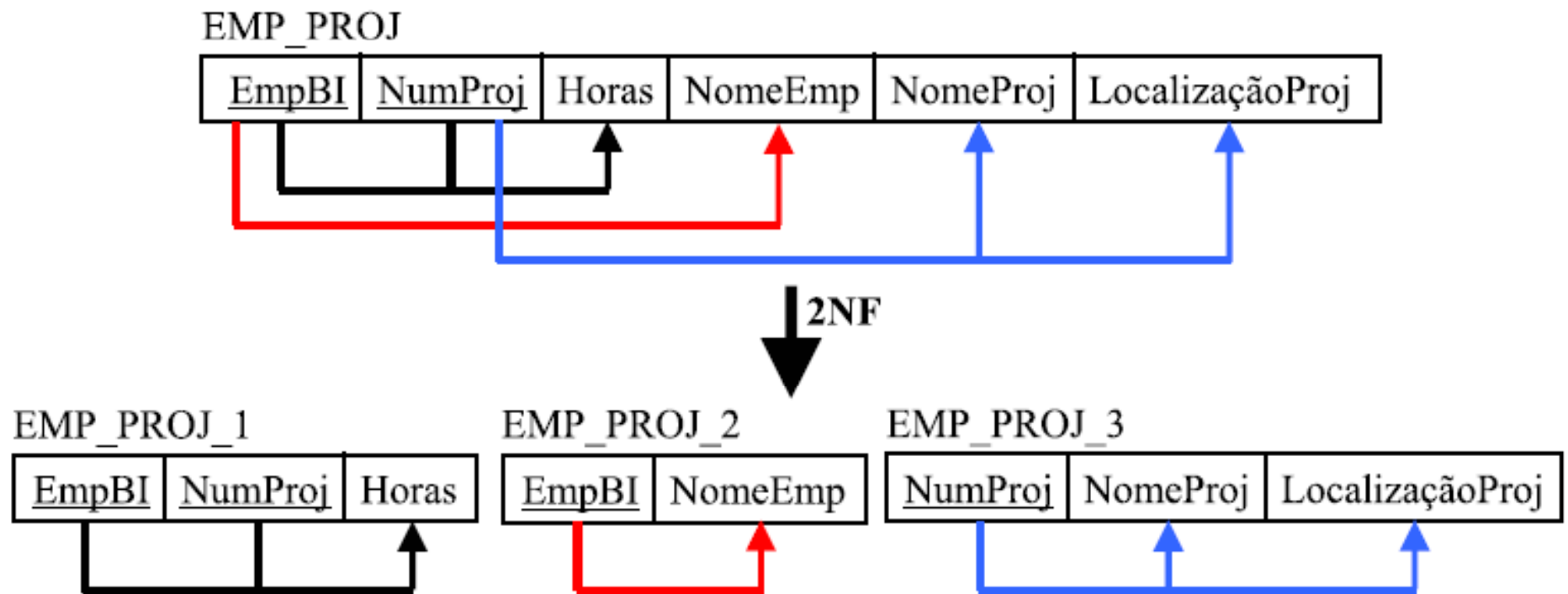
# 2FN – 2ª Forma Normal

- Um esquema relacional está na segunda forma normal se:
  - Está na 1NF.
  - Todos os atributos não-chave dependem por completo da chave primária.
- Na relação EMP\_PROJ, as dependências funcionais  $\text{EmpBI} \rightarrow \text{NomeEmp}$  e  $\text{NumProj} \rightarrow \{\text{NomeProj}, \text{LocalizaçãoProj}\}$  violam a 2NF.



# 2FN – 2ª Forma Normal

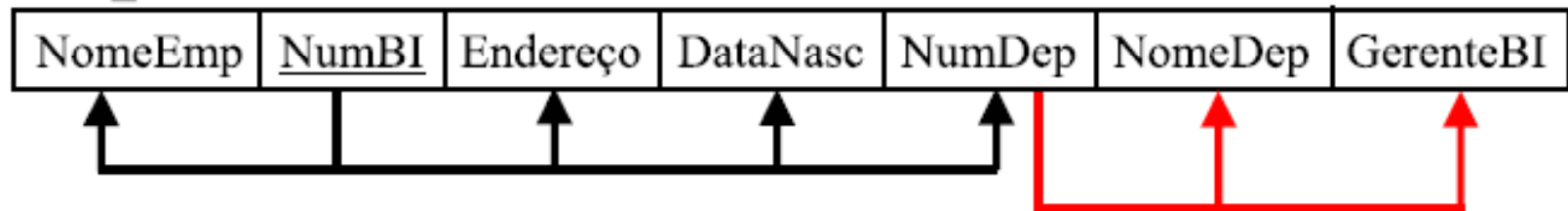
- Decompor em relações com chave externa de forma a que, em cada relação, todos os atributos não-chave dependam por completo da chave primária correspondente.



# 3FN – 3ª Forma Normal

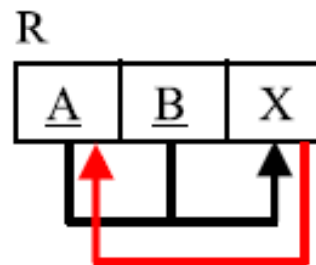
- Um esquema relacional está na terceira forma normal se:
  - Está na 2NF.
  - Nenhum atributo não-chave depende por transitividade da chave primária.
- Na relação EMP\_DEP,  $\text{NumBI} \rightarrow \{\text{NomeDep}, \text{GerenteBI}\}$  é uma dependência transitiva que viola a 3NF.

EMP\_DEP



# Forma Normal de Boyce-Codd (FNBC)

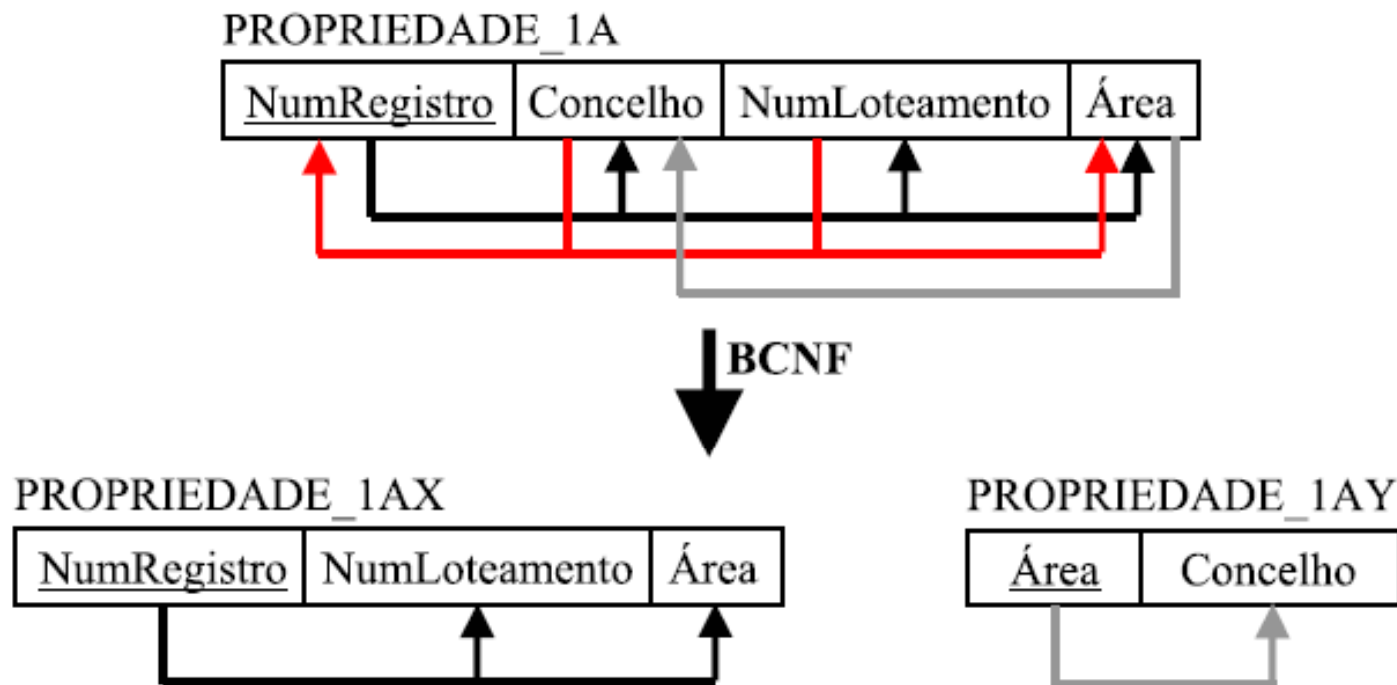
- Um esquema relacional está na forma normal de Boyce–Codd se para qualquer dependência funcional  $X \rightarrow A$  com  $A \notin X$ ,  $X$  é uma superchave.
- A diferença para 3NF é a inexistência da possibilidade de  $A$  ser um atributo de uma qualquer chave. Note-se que se um esquema relacional está na BCNF então também está na 3NF. O inverso pode não ser verdadeiro.
- Na prática, quando um esquema relacional está na 3NF, normalmente também está na BCNF. A exceção é quando existe uma dependência  $X \rightarrow A$  com  $X$  sem ser uma superchave e  $A$  sendo um atributo de uma chave.





# Forma Normal de Boyce-Codd (FNBC)

- A normalização BCNF não verifica a propriedade de preservação das dependências, podendo originar a perda de dependências funcionais.



# Propriedade das Decomposições

- A decomposição é uma ferramenta que nos permite eliminar a redundância, porém tem que observar se não adicional novos problemas
- Propriedades da decomposição:
  - ▣ Decomposição sem Perda de Junção
  - ▣ Decomposição com preservação da dependência
- Note que só é garantido que se preservam as dependências no conjunto de todas as relações da decomposição. Mesmo que haja um subconjunto dessas relações que contenha todos os atributos da relação original, não se garante que as dependências sejam preservadas.

# Decomposição sem Perda de Junção

Seja  $R$  um esquema relacional e  $F$  o conjunto de DFs que valem em  $R$ . Uma decomposição de  $R$  em dois esquemas com conjunto de atributos  $X$  e  $Y$  é dita ser uma decomposição junção-sem-perda em relação a  $F$  se, para toda instância  $r$  de  $R$  que satisfaz as dependências em  $F$ ,  $\pi_X(r) \bowtie \pi_Y(r) = r$ . Em outras palavras, podemos reaver a relação original a partir das relações decompostas.

Se tomarmos projecções da relação e as recombina-mos usando junção natural, e com um obtermos algumas tuplas que não estavam na relação original.

| $S$   | $P$   | $D$   |
|-------|-------|-------|
| $s_1$ | $p_1$ | $d_1$ |
| $s_2$ | $p_2$ | $d_2$ |
| $s_3$ | $p_1$ | $d_3$ |

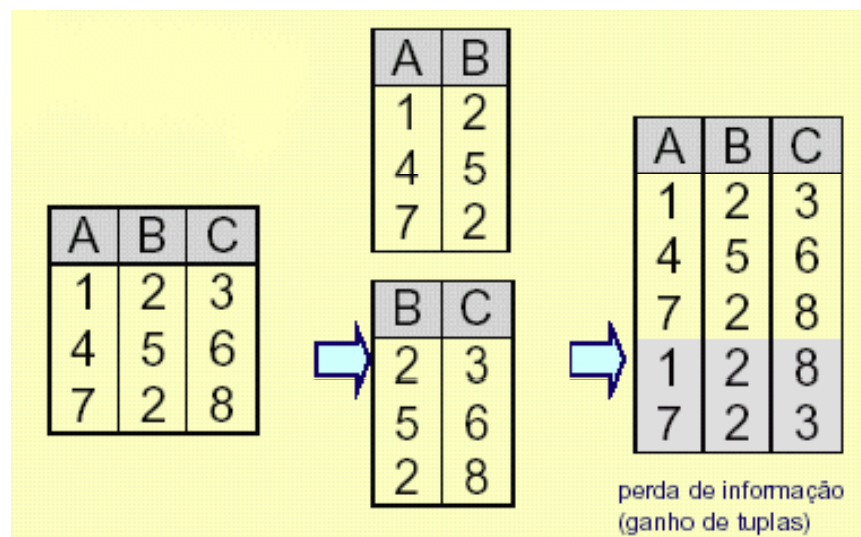
| $S$   | $P$   |
|-------|-------|
| $s_1$ | $p_1$ |
| $s_2$ | $p_2$ |
| $s_3$ | $p_1$ |

| $P$   | $D$   |
|-------|-------|
| $p_1$ | $d_1$ |
| $p_2$ | $d_2$ |
| $p_1$ | $d_3$ |

| $S$   | $P$   | $D$   |
|-------|-------|-------|
| $s_1$ | $p_1$ | $d_1$ |
| $s_2$ | $p_2$ | $d_2$ |
| $s_3$ | $p_1$ | $d_3$ |
| $s_1$ | $p_1$ | $d_3$ |
| $s_3$ | $p_1$ | $d_1$ |

# Decomposição com preservação da dependência

- Uma decomposição é sem perdas se for sempre possível reconstruir a instância da tabela original efetuando a junção das instâncias correspondentes das tabelas decompostas.
- Exemplo de decomposição com perdas:



# Decomposição com preservação da dependência

- A decomposição de um esquema R em sub-esquemas X e Y é sem perdas se e somente se pelo menos uma das duas DFs abaixo valer:
- **X intersecção Y  $\rightarrow$  X, ou X intersecção Y  $\rightarrow$  Y.** Caso especial: se valer a dependência  $U \rightarrow V$ , então a decomposição de R em UV e R - V é sem perdas.

